EXPERIMENTAL STUDIES ON CONTINUOUS SPEECH RECOGNITION USING NEURAL ARCHITECTURES WITH “ADAPTIVE” HIDDEN ACTIVATION FUNCTIONS

Sabato Marco Siniscalchi$^{1,2}$, Torbjørn Svendsen$^1$, Filippo Sorbello$^2$, and Chin-Hui Lee$^3$

$^1$Department of Electronics and Telecommunications, NTNU, Trondheim, Norway
$^2$Dipartimento di Ingegneria Informatica, University of Palermo, Palermo, Italy
$^3$School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA

{marco77, torbjorn}@iet.ntnu.no, sorbello@unipa.it, chl@ece.gatech.edu

ABSTRACT

The choice of hidden non-linearity in a feed-forward multi-layer perceptron (MLP) architecture is crucial to obtain good generalization capability and better performance. Nonetheless, little attention has been paid to this aspect in the ASR field. In this work, we present some initial, yet promising, studies toward improving ASR performance by adopting hidden activation functions that can be automatically learned from the data and change shape during training. This adaptive capability is achieved through the use of orthonormal Hermite polynomials. The “adaptive” MLP is used in two neural architectures that generate phone posterior estimates, namely, a standalone configuration and a hierarchical structure. The posteriors are input to a hybrid phone recognition system with good results on the TIMIT corpus. A scheme for optimizing the contributions of high-accuracy neural architectures is also investigated, resulting in a relative improvement of $\sim 9.0\%$ over a non-optimized combination. Finally, initial experiments on the WSJ Nov92 task show that the proposed technique scales well up to large vocabulary continuous speech recognition (LVCSR) tasks.

Index Terms— Neural networks, Speech recognition.

1. INTRODUCTION

Neural networks (NNs) are powerful pattern recognition tools that have been used for several real world applications over the past years [1], and different successful techniques have been developed around them since the early ‘80s in the speech community as well. For example, in connectionist speech recognition systems [2], NNs are used to estimate the state emission probabilities of a hidden Markov model (HMM) using Bayes’ rule. In the so called Tandem approach [3], a NN is instead used to extract speech features based on phone posterior probabilities on which a conventional HMM-based speech recognizer is trained. In detection-based ASR paradigms (e.g., [4]), a set of NNs is built to learn the mapping from a spectral-based feature space to a phonetic feature space. Although several architectures have been proposed to tackle different speech recognition tasks, such as recurrent neural networks (e.g., [5]), the MLP is by far the most popular due to the compromise realized between recognition rate, recognition speed, and memory resources. Furthermore, it has been shown that feed-forward neural architectures with one hidden layer can approximate various kinds of functions defined on compact sets in $\mathbb{R}^n$ [6], that is, they are universal approximators [1].

Recently, we have seen that high performance speech recognizers can be built by arranging sets of MLPs into multi-stage configurations, e.g., [7, 8, 4]. The leading idea of this approach is to produce a hierarchical integration of phonetic and lexical knowledge during the estimation of the phone posterior probability values. The focus is on the structure of the overall system, so the internal structure of the processing neuron gets less attention. All of the hidden nodes thus share the same structure that is usually based on a sigmoidal activation function. Sigmoidal activation functions are employed because of their nice well-behaved properties and their very simple derivative forms [1]; nevertheless, it has been also reported that they are not suitable for several regression problems [6]. Furthermore, it has never been shown that the use of the same activation function provides optimal generalization capabilities and the best performance, and different scientists in the neural network community have several times mentioned the idea of using different activation functions for different hidden neurons [9]. These ideas have been investigated in the neural network community with beneficial effects on both regression and classification tasks [10, 11]. In [10], the authors observed that Hermite polynomial activation functions enhance the generalization capability of a feed-forward neural architecture. Li et al. [11] showed that a systematic and rigorous methodology for designing constructive feed-forward MLPs can be outlined if each hidden unit employs a Hermite polynomial function as non-linearity. Nonetheless, in both studies the experimental setup is limited to either toy examples, such as the two spirals problem [9], or to the use of limited amount of training data when concerning real-life benchmark problems, for example, only 350 training samples are available in the “Cancer” classification problem [11].

In this paper, we explore the idea of using single-hidden-layer MLPs with different activation functions for different hidden neurons in the setting of continuous phone recognition, where thousands of training data points are available. The activation functions of the proposed architecture are automatically learned from the training data and can change shape during the training phase. This capability is achieved through the use of the Hermite-regression formula as for [10, 11]. We evaluate the goodness of this approach on the TIMIT corpus. Our experiments indicate that this “adaptive” MLP helps reduce the phone error rate (PER) in a stand alone configuration. A PER of 25.0% is achieved by arranging Sigmoid- and Hermite-based MLPs trained on short-time spectral features into a hierarchical structure of NNs. We also linearly combine this hierarchical MLP-based system with our previously proposed detector-based phone recognizer [4], and a PER of 23.0% is achieved with a bigram language model (LM). Finally, a relative improvement of 14.8% is observed in a cross-corpus LVCSR setup when the Hermitian-based NN is used.

The rest of the paper is organized as follows. Section 2 describes the Hermitian-based neural architecture. In Section 3, the hierarchi-
Fig. 1. Neural architecture with Hermitian activation functions.

cal structure of neural architectures is presented. The linear combining procedure is outlined in Section 4. Section 5 describes the experimental results, and Section 6 concludes the paper.

2. SINGLE STAGE NEURAL ARCHITECTURE WITH ADAPTIVE ACTIVATION FUNCTIONS

The neural architecture is a one-hidden-layer feed forward MLP, and is designed for estimating phone posterior probabilities in a discriminative way (see Figure 1). If \( x = (x_1, ..., x_M) \) indicates an input feature vector with \( M \) components, and \( y \in \mathcal{Y} \) is the phone label of \( x \), where \( \mathcal{Y} \) is the set of \( N \) phones (in the experiments we use a set of \( N = 40 \) phones as explained later). The MLP estimates the conditional probability of a phone label \( y \) given an input vector \( x \) using a nonlinear model of the form

\[
\hat{p}_k = \hat{p}(y = k | x) = \frac{\exp g_k}{\sum_{i=1}^{N} \exp g_i},
\]

where \( g_k \) is the linear activation function of the \( k \)th output, and it is given by

\[
g_k = \sum_{j=1}^{L} w_{kj}^{(2)} f_j \left( \sum_{i=1}^{M} w_{ji}^{(1)} x_i \right). \tag{2}
\]

Here \( w_{kj}^{(2)} \) and \( w_{ji}^{(1)} \) denote a weight in the second and first layer, respectively: \( f_j \) is the activation function of the \( j \)th hidden neuron. Either a Sigmoidal or Hermitian activation function can be used in the hidden neuron. If Hermite polynomials are chosen, \( f_j \) is a linear combination of Hermite functions of the form

\[
f_j(z) = \sum_{r=1}^{R} c_r h_r(z), \tag{3}
\]

where \( R \) is the degree of the Hermite polynomial, and \( h_r(z) \) is the \( r \)th Hermite orthonormal function. The orthonormal Hermite polynomials will be described later along with their first-order derivatives.

We want to comment on some of the main differences between the proposed “adaptive” MLP implementation and other similar architectures. In the constructive MLP system [11], the neural architecture grows as part of the training phase by adding new hidden neurons until convergence is reached. Furthermore, the order of the Hermite-based activation function increases by one each time a new hidden unit is added to the network. We believe that this architecture is not suitable for speech applications where the number of hidden neurons is on the order of hundreds or thousands, and it did not seem reasonable to use Hermite polynomials with such a high degree. In our implementation the number of hidden neurons is therefore fixed at the beginning of the training phase. All of the hidden neurons employ a Hermite polynomial of the same degree, which can be selected between one (1) and fourteen (14). This results in a configuration more similar to [10], but there linear activation functions are used at the output layer; moreover, the sum-of-squares is chosen as error function to be minimized. In contrast, the softmax function is employed in this study, and the cross-entropy error function is chosen as function to be minimized during the training phase.

2.1. Hermite Regression Formula

The orthonormal Hermite polynomials, \( H_r(z) \), are defined over the interval \( (-\infty, \infty) \), and there exist several ways to formally introduce them; in this study, we follow [10], and the orthonormal Hermite function of order \( r \) is then given by

\[
h_r(z) = \alpha_r H_r(z) \phi(z), \tag{4}
\]

where

\[
\alpha_r = (r!)^{-\frac{1}{2}} \frac{1}{\sqrt{2\pi}} \frac{1}{(2^r) \Gamma \left( \frac{r}{2} \right)}, \tag{5}
\]

\[
\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}, \tag{6}
\]

\[
H_r(z) = (-1)^r e^{-z^2} \frac{d^r}{dz^r} \left( e^{-z^2} \right) = 2z H_{r-1}(z) - 2(r-1) H_{r-2}(z), \tag{7}
\]

\( r > 1, \quad H_0(z) = 1, \quad H_1(z) = 2z. \)

The recursive nature of the Hermite polynomials makes the computation of the first-order derivative of Eq. 3 very simple and therefore its use as activation function is quite appealing. The first-order derivative is given by

\[
\frac{d}{dz} f(z) = \sum_{r=1}^{R} c_r \frac{d}{dz} (h_r(z)) = \sum_{r=1}^{R} c_r \left[ (2r)^{\frac{3}{2}} h_{r-1}(z) - z h_r(z) \right]. \tag{8}
\]

Fig. 2. Two-stage neural architecture.
3. TWO STAGE APPROACH FOR PHONE POSTERIOR ESTIMATION

The single-stage approach described in Section 2 uses only spectral and limited temporal information (delta and delta-delta feature) to discriminatively estimate phone posteriors, thereby disregarding additional sources of information, such as phonetic and lexical knowledge, that can help generate better phone posterior estimates. Indeed, better results, in terms of posterior estimates, have been reported by taking into account long temporal context information [7, 8, 4, 12], which in turn usually leads to better ASR performance. Generally speaking, long temporal dependencies can be thought of as lexical and/or phonetic knowledge. A common approach to integrating this additional information with what is used in common neural architectures is to build a hierarchical structure of NNs. Only two layers are usually employed, and long temporal context is exploited either at the input of the first layer [7, 8, 4], or at the input of the second layer [12].

In this section, we propose a new neural architecture based on a) an ensemble of MLPs with different hidden activation functions, and b) a non-linear two-stage configuration. The key idea is to hierarchically integrate phonetic and lexical knowledge during the phone posterior estimation. Figure 2 shows the proposed solution. The MLPs in the first layer are trained on short-time spectral features and learn the mapping from the acoustic to the phone space. Two MLPs with different hidden activation functions are used in our experiments, namely, Hermite polynomials and Sigmoids. A single MLP is used in the second layer which takes long temporal context of the posteriors estimated by the first-stage MLPs as its input and generates phone-state posterior estimates (in the experiments we use \( S = 3 \) states per phone, as explained later). The goal of this stage is two-fold: a) to merge the output of the MLPs in the first layer, and b) to learn long term dependencies between phone evidences.

It may be worth to point out that our approach is similar to [12], but there are several differences. For example, two MLPs are used in the first layer in our study instead of one; the goal of the second layer is not only to exploit long temporal context but also to merge the output of different neural architectures; and phone-state posteriors are produced by our configuration rather than phone posteriors.

4. LINEAR COMBINATION SCHEME

Combination schemes of local phone posteriors have long been used in speech recognition (e.g., [13]). When the neural architecture is trained such that its output can be interpreted as a Bayesian posterior probability, several linear combination schemes can be used, e.g., product rule and sum rule.

In this work, a slightly modified version of the linear combination rule is adopted as a way to combine the output of two phone classifiers. Given two (2) neural classifiers and \( V \) classes \( c_1, \ldots, c_V \), the sum rule computes

\[
P(c_k|x) = \alpha P^{(1)}(c_k|x) + \beta P^{(2)}(c_k|x), \quad k = 1, \ldots, V
\]

\[
\alpha + \beta = 1, \quad \alpha \geq 0, \quad \beta \geq 0.
\]

(9)

Here \( \alpha \) and \( \beta \) denote the interpolation coefficients, and \( P^{(i)}(c_k|x) \) is the phone posterior probability for class \( c_k \) estimated by the \( i \)th classifier. Note that if \( \alpha = 0 \) the final score is estimated using the second classifiers only; whereas, only the first classifiers is used if \( \beta = 0 \). In Eq. 9, all of the involved posterior probabilities are frame-wise, and equal prior probabilities is assumed.

5. EXPERIMENTS

In the following sections we present the experimental setup and discuss the results.

5.1. Phone Recognition Experiments

All experiments were conducted using the TIMIT corpus [14]. We mapped the original TIMIT broad-phonetic labels into 39 phones as prescribed in [8]. Mel-frequency Cepstral Coefficients (MFCCs) were used as acoustic parametrization of the speech signal. The first and second time derivatives of the cepstra were computed as well and appended to the static cepstra to yield a 39-dimensional feature vector. Training and test data were generated using the available hand-labeled phone time-stamps. The TIMIT material was split into training (3296 sentences), development (400 sentences), and evaluation (1344 sentences) sets, and the classification accuracy is given for 40 classes (39 phone classes, and 1 garbage class). All of the NNs are feed-forward one-hidden-layer MLPs.

In the experiments with single-stage MLPs two neural architectures were designed: one with Hermite hidden activation functions, and another with Sigmoidal hidden activation function. Both MLPs implement a mapping from the acoustic to the phone space. Hence, the number of inputs of each MLP is equal to 39 (i.e., equal to the number of component of the MFCC vector). The number of outputs is 40, which corresponds to the number of phones plus the garbage class. The softmax non-linearity is used at the output layer. The standard back propagation algorithm was used as training method, and to avoid over-fitting the reduction in classification error on the TIMIT development set during the training phase was adopted as stopping criterion. The number of nodes in the hidden layer was set equal to 500. The degree of the Hermite polynomial was set equal to 14. In the experiments conducted with the system shown in Figure 2, three NNs were used. The two MLPs in the first layer are actually the ones of the previous experiment. The third MLP acts as a merger and implicitly integrates lexical and phonetic knowledge during the phone posterior estimation phase. This MLP is thus trained using the concatenated output of the MLPs in the first layer along with a temporal context spanning 13 frames. The number of inputs is therefore equal to 1040, the number of outputs is 120 (each phone has been split into three states), and 1500 is the number of hidden neurons with Sigmoidal activation function. The softmax function is used at the output layer. This MLP is implemented with the Quicknet\(^1\) toolkit.

Table 1 summarizes the frame accuracy rates (FARs) and the relative improvement of the Hermitian-based MLP (HMLP) over the standard Sigmoidal-based MLP (SMLP) in the single-stage configuration experiments. The HMLP outperforms the SMLP on both data-sets achieving a 2.8% and 2.4% relative improvement for the validation and evaluation sets, respectively. Continuous phone recognition is carried out using the phone posterior estimates in a hybrid NN/HMM system, that is, as local scores for decoding. PERs obtained with a 0-gram language model and with no realignment are shown in Table 2. The FARs are confirmed in the continuous phone recognition task, and the HMLP system outperforms the SMLP system. Specifically, a 5.5% relative improvement is reported on the evaluation set (see last row of Table 2).

The above results are promising, yet very far from the \(~25.0%\) PER recently reported in [8, 4]. To bridge this gap, we use the two-stage architecture presented in Section 3 with the configuration already described in Section 5.1. Looking at Table 3, we can see that PERs of 25.7% and 25.0% can be achieved with a 0-gram and b-

\(^1\)SPRACHcore package, http://www.icsi.berkeley.edu/~dpwe/projects/sprach/
Table 1. Frame accuracy rates (FARs) for single-stage neural architecture on TIMIT.

<table>
<thead>
<tr>
<th>System/Dataset</th>
<th>Validation</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMLP</td>
<td>61.84%</td>
<td>60.09%</td>
</tr>
<tr>
<td>HMLP</td>
<td>63.57%</td>
<td>61.54%</td>
</tr>
<tr>
<td>Rel. Improv.</td>
<td>2.80%</td>
<td>2.40%</td>
</tr>
</tbody>
</table>

Table 2. PERs for single-stage neural architecture on TIMIT. A 0-gram language model is used.

<table>
<thead>
<tr>
<th>System/Dataset</th>
<th>Validation</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMLP/HMM</td>
<td>38.22%</td>
<td>38.95%</td>
</tr>
<tr>
<td>HMLP/HMM</td>
<td>35.74%</td>
<td>36.79%</td>
</tr>
<tr>
<td>Rel. Improv.</td>
<td>6.5%</td>
<td>5.5%</td>
</tr>
</tbody>
</table>

gram LM, respectively. It is also important to point out that these low PERs are achieved using short-time frequency features as input vectors rather than long-temporal features [8, 4].

Finally, we use Eq. 9 to enhance posterior estimates and thus improve phone recognition rates. In particular, we linearly combine our detector-based phone recognition system [4] and the two-stage system used in the previous experiment. For the evaluation set, the combined system achieves a PER equal to 23.65% with a 0-gram LM. The PER is further reduced to 23% with a bigram LM, which represents a relative improvement of ~9% over both the two-stage and the detector-based recognition systems. The validation set was used to tune α and β.

5.2. Word Recognition Experiments

The re-descending shape of the Hermite polynomials reduces the size of the saturation zones, and that may reveal to be a key element in paradigms such as TANDEM or BOTTLENECK. To this end, we report a preliminary LVCSR experiment using the MLP posterior probabilities as features to train a TANDEM system. We address the speaker independent Wall Street Journal (5k-WSJ0) task. Due to time constraints, the SMLP and HMLP systems presented in Section 5.1 are not trained on WSJ0-specific material, and a cross-corpus setup is preferred. Posterior features are therefore extracted by forwarding the SI-84 training material (7077 utterances from 84 speakers, i.e., 15.3 hours of speech material) and the Nov92 evaluation material (330 utterances from 8 speakers) through the TIMIT-trained networks. The posterior features are in turn used to MLE train two TANDEM systems, and we refer to them as WSJ-SMLP and WSJ-HMLP, respectively. Table 4 shows word error rates (WERs) when 8 Gaussian mixture components per state and a trigram LM are used. The WSJ-HMLP system outperforms the WSJ-SMLP system on this LVCSR task as well, demonstrating that the proposed technique scales up to more difficult tasks than phone recognition and works better than conventional NN in cross-corpus conditions. Although state-of-the-art HMM/GMM system trained and tested on WSJ0 specific material can achieve lower WERs, it is interesting to notice that results qualitatively similar to ours using a TANDEM system trained and tested on 5k-WSJ0 material were reported in [15] (see last column of Table 4). We therefore expect to achieve a better performance training on the WSJ0 material.

6. CONCLUSIONS

A relatively novel neural architecture which has the capability to change the shape of its hidden neurons during the training phase through the use of a weighted sum of R orthonormal Hermite functions has been described, and experimental results demonstrate that performance is always enhanced on both classification and recognition tasks. Moreover, improvement was observed in both matched and mismatched condition (see 5.2), and in both phone and word recognition experiments. Finally, it is not a goal of this work to advocate the superiority of the Hermitian neuron over the Sigmodial one, which must be verified case by case, but it aims at hopefully triggering some new studies on the use of novel neural architectures for speech tasks. Indeed, this line of research is being furthered by training the Hermitian-based system on WSJ0-specific data.

7. REFERENCES